OCCLUSION TOLERANT OBJECT RECOGNITION METHODS FOR VIDEO SURVEILLANCE AND TRACKING OF MOVING CIVILIAN VEHICLES

Nishikanta Pati

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2007

APPROVED:

Partha Sarathy Guturu, Major Professor
Saraju P. Mohanty, Co-Major Professor
Bill P. Buckles, Committee Member
Xiaohui Yuan, Committee Member
Armin Mikler, Program Coordinator
Krishna Kavi, Chair of the Department of Computer Science and Engineering
Oscar Garcia, Dean of College of Engineering
Sandra L. Terrell, Dean of the Robert B. Toulouse School of Graduate Studies

Pati, Nishikanta. Occlusion Tolerant Object Recognition Methods for Video Surveillance and Tracking of Moving Civilian Vehicles. Master of Science (Computer Engineering), December 2007, 52 pp., 1 table, 9 figures, 38 titles.

Recently, there is a great interest in moving object tracking in the fields of security and surveillance. Object recognition under partial occlusion is the core of any object tracking system. This thesis presents an automatic and real-time color object-recognition system which is not only robust but also occlusion tolerant. The intended use of the system is to recognize and track external vehicles entered inside a secured area like a school campus or any army base. Statistical morphological skeleton is used to represent the visible shape of the vehicle. Simple curve matching and different feature based matching techniques are used to recognize the segmented vehicle. Features of the vehicle are extracted upon entering the secured area. The vehicle is recognized from either a digital video frame or a static digital image when needed. The recognition engine will help the design of a high performance tracking system meant for remote video surveillance.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

We are blessed with light from the beginning. Either visible or invisible - it illuminates and saturates the world. Technically speaking we don't see light, but see objects with it. In the past, ancient Greeks thought that our eyes are like lanterns, sending out light rays that made objects visible when struck. This concept was undoubtedly held for more than 15 centuries until Arab scholar Al-Hazen about A.D. 1000 produced arguments to prove this concept wrong. Visual information reaching us by reflections of light plays a vital role in our ability to interact with the world more interactively. Human eyes are probably the one of the most amazing creations which provide color and 3-D representations of scenes within moments. Without the ability to process visual information, we would be severely handicapped and it is therefore not surprising that major half of the primary cortex is devoted to visual information processing.

Technology revolution has enabled availability of more affordable digital video acquisition devices in the market. This has demanded the attention of researchers and software developers to build more applications for digital video. The tremendous success of Web camera applications and the usage of high definition digital video cameras everywhere, we believe that use of digital video will soon become a reliable technique for security and surveillance planning. Unlike still images, video sequences provide more information about the movements of the objects and scenarios change over time, but at the cost of increased space for storage and wider bandwidth for transmission. However, the extra information which comes at an additional cost lays the stepping stone for building many applications automating thousands of process and enhancing the quality of the process.

Real-time object tracking has become an essential tool for many processes such as security monitor, surveillance, perceptual user interfaces, smart rooms, object-based video compression, and driver assistance. It has been extensively used by aircraft control stations since decades. A typical visual tracking system consists of two major components: Object localization and representation. This system should also be responsive toward the changes in the appearance of the target. Tracked object segmentation and feature data association is mostly a top-down process dealing with the dynamics of the tracked object. The way the two major components are combined and weighted is application dependent and plays an important role in the robustness and efficiency of the tracker. For example, people tracking in a crowded scene concentrates more on target (face) representation than on the dynamics of the target. On the other hand, aerial video surveillance focuses on the target movement and motion of the camera plays important role. But, overall expectation of all the visual tracking systems remain the same. The system should be smart enough to learn the changes in the background scenes quickly and thereby updating the background model. Ideal system should be smart enough to evaluate different hypotheses on the run to change the system settings dynamically to produce superior performance. In certain real-time applications, we don't have the liberty to use large fraction of the system resources for tracking. The majority of resources are used for the preprocessing stages or to high-level tasks such as recognition, trajectory interpretation, and reasoning. Therefore, it is highly essential to keep the computational complexity of a tracker as low as possible.

1.1. Motivation Behind Video Object Tracking

Adequate visual information is available in digital form inside a digital video or static images captured in periodic intervals. The emergence of easy availability digital video cameras and its usage in everyday's life from simple video capture for entertainment to recording evidences used for court proceedings, has demanded the attention of researchers and software developers to leverage the rich information available in

a digital video and build software applications to automate many processes. Main purpose of video segmentation is to enable content-based representation by extracting objects of interest from a series of consecutive video frames. Briefly, motivation behind video object tracking is to harness the power which enables several important applications such as: Security and surveillance - to recognize people, to provide better sense of security using visual information; Medical therapy - to improve the quality of life for physical therapy patients and disabled people; Retail space instrumentation - to analyze shopping behavior of customers, to enhance building and environment design; Video abstraction - to obtain automatic annotation of videos, to generate object-based summaries; Traffic management - to analyze flow, to detect accidents; Video editing - to eliminate cumbersome human-operator interaction, to design futuristic video effects; Interactive games - to provide natural ways of interaction with intelligent systems such as weightless remote control. Some sample uses of video object tracking are illustrated in Figure 1.1.

1.2. Problem Definition

A digital video-based vehicle tracking system detects and tracks individual vehicle that is moving through the camera scene. This system can provide location and hereabouts of a vehicle entered through the gate. The system provides not only basic traffic parameters, including the vehicle count, availability of parking space, but also traffic flows such as normal or slow traveling of vehicles, vehicle traveling in the wrong direction, and stopped vehicles on narrow streets. Recently, digital video-based traffic surveillance and security enforcement becomes an important topic in the intelligent traffic system (ITS). Moving vehicles are detected and tracked automatically in monocular image sequences from road traffic scenes recorded by a stationary camera. In order to exploit the a priori knowledge about shape and motion of vehicles in traffic scenes, a parameterized vehicle model is used for an intra frame matching process and an adaptive background estimator is used to model the background from time to time. Initially, the representative feature parameters of a newly entered vehicle are

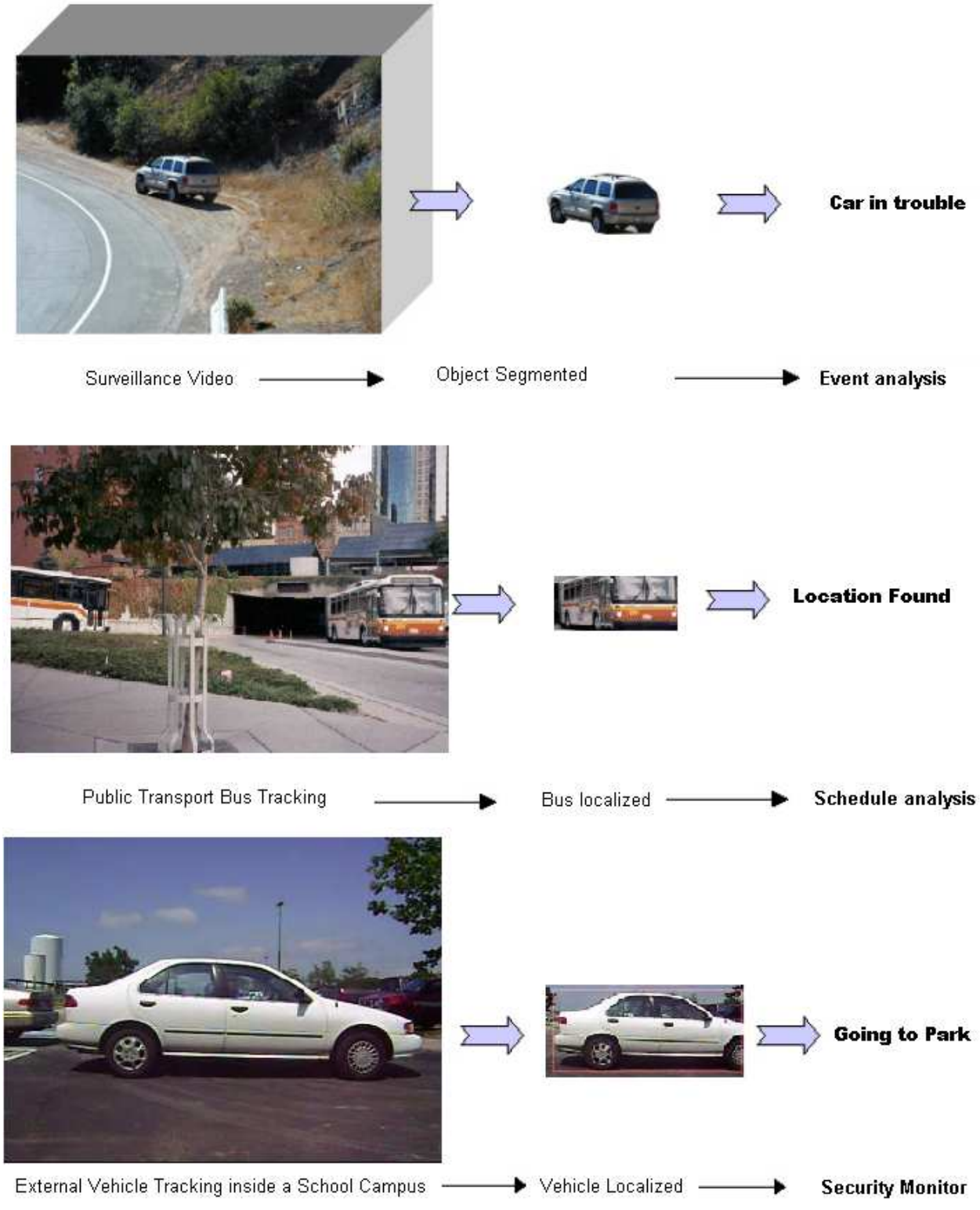


FIGURE 1.1. Sample Uses of Real-time Video Object Tracking

extracted using an image segmentation module. The system tags the vehicle with a tracking number and stores the extracted parameters used to represent the vehicle in the database. Later, when a vehicle is found in a video frame or image scene, similar procedure is followed to segment the object and extract candidate representations of vehicle considered as a video object or image of a moving vehicle. The, the system tries to find the best possible match of the vehicle with all the registered vehicle and updates the monitor screen with the current details of the recognized or classified vehicle. The inclusion of a dynamically updating background model and shadow removal module allows to remove shadow edges of the vehicle and get an enhanced image of the vehicle from the static image or video captured recently. An elaborate combination of various techniques has enabled us to track vehicles under complex illumination conditions and often obscured behind other vehicles or objects. Figure 1.2 shows the schematic architectural representation of the system. Results on various real world road traffic scenes are presented and open problems as well as future work are outlined.

1.3. Object Recognition Methodology

Object recognition is the core of any object tracking system. The major components in any video-based object recognition system are frame extraction, video object segmentation, and feature based classification. Each of these steps carries equally important value to build a video object recognition system with superior performance. The proposed vehicle recognition system has four main components:

(1) Scene Change Detection or Background Subtraction

(2) Object Segmentation

(3) Feature Extraction

(4) Recognition and Classification

Accurate and reliable object segmentation and recognition under the constraint of partial occlusion and low computational complexity presents a challenge. The goal of

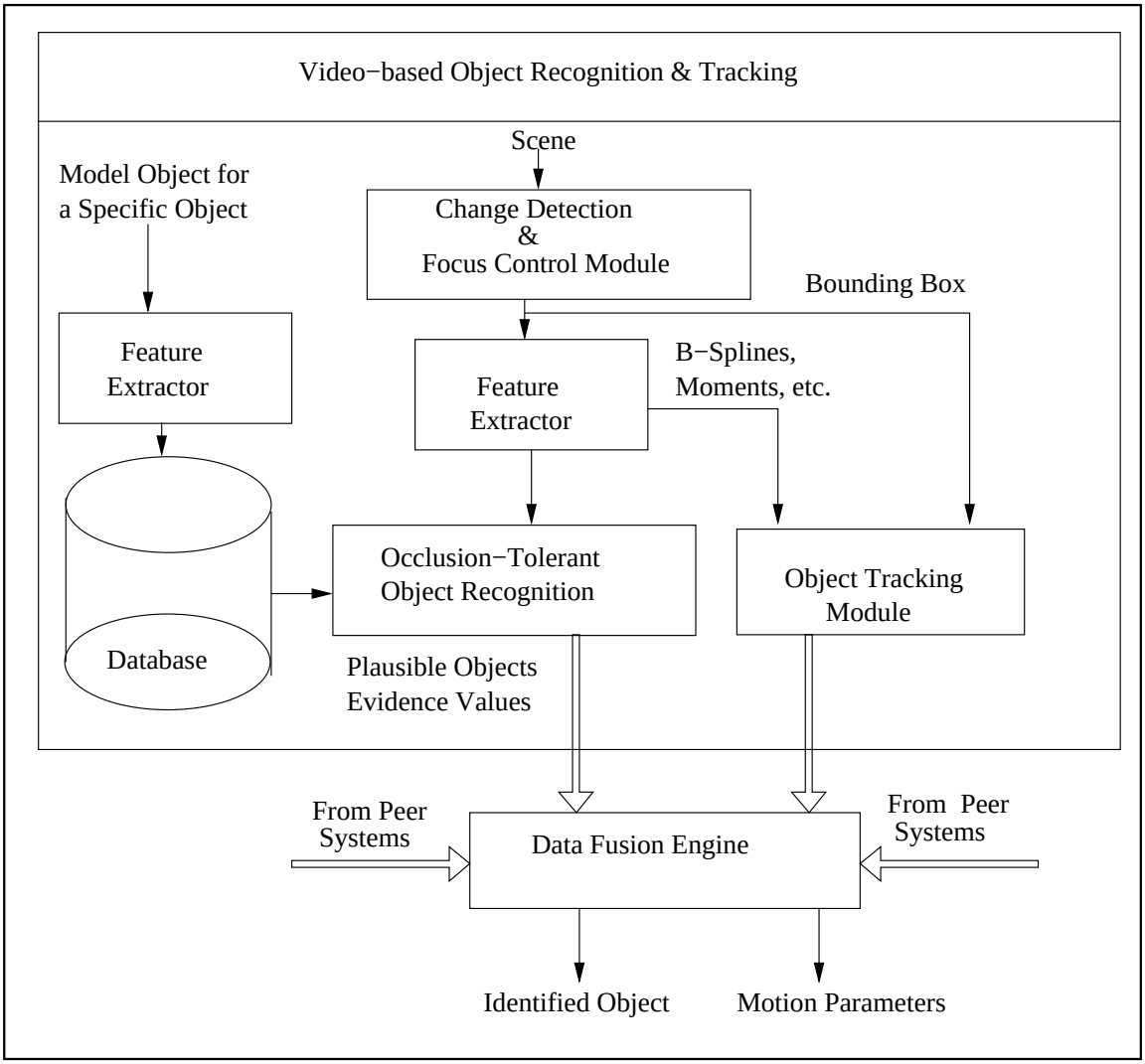


FIGURE 1.2. Overall System Architecture for Moving Vehicle Tracking

this work is to find and develop optimized algorithms that are robust, noise-tolerant, simple, modular, and easily adaptable to various applications.

In order to detect and analyze non-trivial events in road traffic scenes, we have to consider the following dilemma: Either we have to fix the camera on an interesting agent by applying gaze control so that the tracked object remains in the field of view. Or we must use a stationary camera with a field of view that is large enough to capture significant actions of moving agents. Another solution to this problem is to install multiple low-cost, small field-of-view cameras and use a fusion algorithm to provide required scene data to the object tracking system. Additionally, in a

real-time road traffic scene, we have to cope with a cluttered environment full of background features as well as with occlusions and dis-occlusions. This makes the task of foreground-background segmentation extremely difficult. The appropriate use of models for background estimation and subtraction from the current image or video frame plays vital role in successful segmentation of foreground objects intended for recognition.

Object tracking techniques can be broadly divided into two categories: recognition-based tracking and motion-based tracking. Recognition-based tracking uses the concept of recognition and classification of the object in successive images and the extraction of its location and prediction of activities from the movements. The main advantage of this tracking method is that it can be accomplished in three dimensions, and that the object translation, rotation and scale can be estimated. The disadvantage is that only pre-defined objects can be tracked, and, as a result, the tracking performances are limited by the high computational complexity of the recognition strategy. Motion-based object tracking systems depend entirely on motion parameter estimation to detect the object. They have the added advantage of being able to track any moving object independent of size or shape of the object.

1.4. Background Subtraction

The first step in any moving object tracking system, which distinguishes moving objects from the stationary background. I apply a simple pixel based differencing to detect changes happened in the scene. But, the disadvantage of this approach is that some background objects are often detected as foreground objects because of the changes in illumination, movement of leaves and presence of walking people. To compensate for this problem, I have a module which dynamically updates the background model to handle the changes in the background scene mentioned above. Apart from the changes in the background, cast shadows around the object due to direct blocking of the light adversely affect the object recognition accuracy. A shadow removal algorithm has been applied to remove the shadowed edge of the

segmented object. The identification of coherently moving image features provides a rough estimate of moving regions in the image.

1.5. Object Segmentation

Reliable object segmentation plays an important role for object recognition and classification. Improper segmentation of the object from the image or the input video frame adversely affects the performance of of object tracking system. So, suitable methods are adapted for segmenting the objects while ignoring the noise and undesired components. The moving objects are binarized by using a suitable threshold estimated using the local statistics of the image. Blob analysis has been used for initial segmentation and filtering of object blobs. Heuristic based rules are applied to refine the selection of foreground objects. The segmentation module carefully analyzes the features of all the objects present in the scene and filters the objects who satisfy certain criteria.

1.6. Vehicle Model

This step is usually called feature extraction in traditional pattern recognition systems. To recognize and classify an object in an image, We must first extract some features out of the image. Feature extraction is the technique to extract various image attributes for identifying or interpreting meaningful physical objects from images. The primary idea is to represent the visual appearance of an object by distinctive key features, or attributes. The objective of this step is to identify the most discriminative image features with the lowest dimensionality in order to reduce the computational complexity and improve the tracking accuracy.

1.7. Object Recognition and Classification

This is the final and most important step of object recognition process. This is initialized by formulating a model hypothesis using a reference model and initial values of each independent object are extracted during the object registration process. Since model-based tracking depends on object recognition and location extraction,

classification accuracy plays a vital role. During classification, the system generates a feature vector $X = [X1, ..., Xn]$ that represents the encoding of the object in feature space. For example, it can be obtained by measuring the presence of specific visual features of the object. The different features considered in this work are invariant geometric moments, B-spline coefficients of curve segments, stereo disparity parameters, etc. Accommodations have been made to correctly identify objects under partial occlusion. Final decision about the name of the object is performed by plugging the feature vector into a classification function $f(X)$ that returns 1 or 0, depending on whether the presented object is present in the database or not. If the object is found to be registered with the system, the location information is extracted and the position the object is updated on the tracking map.

1.8. Object Location Extraction

In order to track the object under surveillance, the absolute location of the recognized object must be estimated. Object location is detected by a hierarchical matching of the background scene with the complete map of the campus or area. The details of this step is beyond the scope of this thesis.

1.9. Main Contributions

This work is focused on designing and implementing a computationally inexpensive robust and occlusion-tolerant recognition methods for moving vehicle tracking system keeping security surveillance in mind. So, all the important stepping stones discussed in the previous sections are reviewed carefully. Robust yet simple algorithms are chosen to reduce the computational cost. This thesis has addressed a number of challenging issues associated with object segmentation when the quality of the image is poor and noisy. The selection of features for better recognition accuracy are obtained mostly from experiments. So, to summarize selection of appropriate algorithms and representative features via experiments is the main goal of this work.

1.10. Outline of the Thesis

In the next chapter, the existing methodologies and different algorithms proposed by different authors in recent years related to the video object tracking such as region are discussed. Chapter 3 discusses the background modeling and subtraction, shadow removal, and vehicle segmentation step of the project. Chapter 4 outlines the different feature sets used for object recognition and the strategy used to achieve correct identification of objects under partial occlusion. Chapter 5 contains information about the experimental setup and discusses the recognition accuracy obtained from different combinations of feature sets and classifiers. Chapter 6 draws the conclusion of the project and outlines the future directions to the research attempt.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Object recognition deals with the recognition or classification of different objects in two or three dimensional images as instances of predetermined object classes. Object recognition is one of the most important for image analysis and understanding. Object shape has been used as one of the most powerful features to recognize the object [9]. Apart from the shape there are other features which are used frequently like color [19], texture [19], invariant moments [28], depth, topology, etc. that are derived from static two dimensional images. More complex systems use Bayesian and aspect graph methods for robustness and accuracy. Image data often comes with noise, is cluttered along with several different objects. Sometimes the target objects may be occluded or hidden so only a fraction of the object is visible. Apart from the occlusion problem, the object may be present in any location in the image, orientation could be changed and may be scaled in the image. Different parts of the image may be illuminated differently and by different light sources. So, the color and texture of the object may be different in different parts of the same image or in subsequent frames of the video. Keeping all these variation and a multitude of possibilities in mind, the recognition systems are subject to have high computational complexity, long decision latency and are vulnerable to error. Ideally speaking, the object recognition and classification systems should have translation, rotation and scale invariant algorithms. The system must be robust to occlusions, noise and illumination differences. The recognition and classification process should be fast enough to be useful for use for real time applications like surveillance. The system should have the power to use specialized

algorithms running in highly parallel manner for detecting specific features needed to enhance the performance of the system.

Researchers have put a lot of effort to develop robust and sensitive object recognition systems. Several shape based techniques are used for object recognition. Hahnel *et al.* [19] combines shape and color information for object recognition. Worthington *et al.* [36] uses shape-from-shading techniques. Similarly, shape-from-texture methods are used by authors in [19]. Belongie *et al.* [22] uses shape contexts for shape matching. Wavelet transforms for shape matching are used for object recognition by authors in [2]. Pansang *et al.* [26] discussed the most basic shape features for two dimensional image object recognition. Gerard *et al.* [23] have described the use of cubic B-splines for 2-D shape representation and classification. Authors in [18] have shown a way to segment the curve which could be used as the base for segmenting the shape curve of an object for occlusion tolerant object recognition. Raymond *et al.* [34] have explained that the spatial differences between the images seen by the two human eyes, called binocular disparities, can be used for occlusion tolerant object recognition.

2.2. Video Object Tracking System

Image sequence or video frames analysis provides intermediate results for a conceptual description of actions and events in a scene. A system that establishes such higher level descriptions based on tracking of moving objects in the image domain has been described in [20]. Yilmaz *et al.* [38] reviewed the state-of-the-art tracking methods, classified them into different categories, and identified new trends. Chang *et al.* [5] demonstrated an automatic video region tracking and a robust moving objects detection system.

2.3. Shape Based Recognition from Static Images

I have considered the simple shape and contour based features that are used for 2D objects. Moments of different orders are used for classification objects without

occlusion. Boundary length, area and moment of inertia values are included for broad classification. The compactness of the object is determined from the area and boundary length. The area and boundary length are shape and rotation invariant but not scale invariant. Moreover,Hence it would be necessary to normalize the image if these properties are to be used. The moment of inertia about the x, y and xy is also a useful property. To re-orient the object to a standard, the object can be rotated using the axis of minimum moment of inertia about xy. This can be done if the object is not symmetrical or if the ambiguities regarding more than 1 minimum values can be resolved. Moment invariants are very useful to infer the equivalent ellipse, i.e., the best fitting ellipse for a target 2D object. Approximations of the object by a set of ellipses at different levels are used for feature vector construction [28]. Cubic B-splines [3] can represent the shape of the object and after standardization of the object i.e. rotating and shifting to the origin, the coefficients can be used as shape properties. The depth estimations at critical corner points using stereo disparity techniques are also used as explained authors in [34].

CHAPTER 3

BACKGROUND SUBTRACTION AND OBJECT SEGMENTATION

3.1. Introduction

Object segmentation can be accomplished by building a representation of the background scene named as background model and then detecting the changes in each new frame from the model. Any significant change in an image region from the estimated background model signifies a moving object. The areas of the image plane where there is a significant difference between the observed and background model images indicate the location of the moving objects or presence of new object. Usually, a connected component algorithm or blob analysis is applied to obtain connected regions corresponding to the objects. This process is commonly referred to as the background subtraction. Background subtraction is usually the first step for segmenting out objects of interest in a scene for almost all computer vision applications such as video surveillance systems, traffic monitoring, environment monitoring, obstacle detection, etc. The name "background subtraction" comes from the simple technique of subtracting the observed image from the background image and thresholding the result to find the objects on interest. As a matter of fact, this process is also called "scene change detection" as it detects the changes in the original background scene.

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background

objects such as moving leaves, rain, snow, and shadows cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of vehicles.

There are several problems that a good background subtraction algorithm must solve correctly. Consider a video sequence from a stationary camera overlooking a traffic intersection. As it is an outdoor environment, a background subtraction algorithm should adapt to various levels of illumination at different times of the day and handle adverse weather condition such as fog or snow that modifies the background. Changing shadow, cast by moving objects, should be removed so that consistent features can be extracted from the objects in subsequent processing. This process is called shadow_removal. As Cheung *et al.* [30] described that the complex traffic flow at a 4-way intersection also poses challenges to a background subtraction algorithm. The vehicles move at a normal speed when the light is green, but come to a stop when the traffic signal turns red. The vehicles then remain stationary until the light turns green again. A good background subtraction algorithm must be robust enough to handle the moving objects that first merge into the background and then become foreground at a later time. In addition, to accommodate the real-time needs of many applications, a background subtraction algorithm must be computationally inexpensive and have low memory requirements, while still being able to accurately separate foreground objects in the video.

This experiment and research began with a comparison of various background subtraction algorithms for detecting the presence of moving vehicles and objects in a video sequence captured near a parking lot. I considered approaches varying from simple techniques such as frame differencing and adaptive median filtering, to more sophisticated probabilistic modeling techniques. While complicated techniques often produce superior performance, my experiments show that simple techniques such as adaptive median filtering can produce good results with much lower computational complexity.

In addition, I found that pre-and post-processing of the video might be necessary to improve the detection of moving objects.

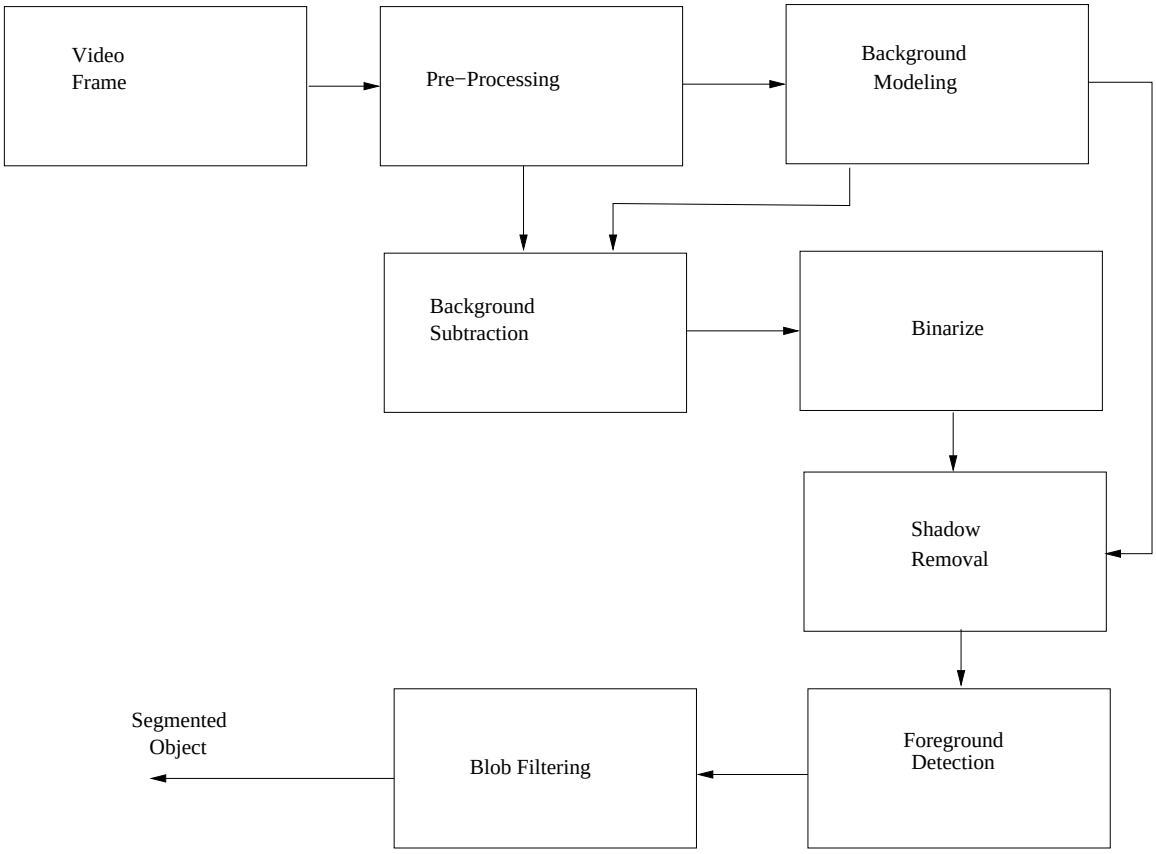


FIGURE 3.1. Architecture of Background Subtraction and Object Segmentation

3.2. Background Subtraction Methods

Even though there exist innumerable background subtraction algorithms in the literature, Background subtraction or Change detection methods can be broadly classified into two categories. The first method is a pixel-based method and the other a region-based method. The simplest method to detect changes is intuitively differencing intensities of corresponding pixels in the grayscale images. If the difference in gray scale exceeds a predefined threshold, the pixel is regarded to be a changed one and considered to be a part of the foreground object. Hence, the determination of the threshold is extremely critical; A high threshold value will suppress significant changes, if the threshold is low it will drench the difference map with spurious changes

and contains a lot of noise. Rosin [29] explained four different pixel-based methods: the noise intensity model, the signal intensity model, the noise spatial model, and the signal spatial model. The models were developed to decide the thresholds and were compared with each other. Wren *et al.* [37] proposed a robust method to select thresholds at each pixel adaptively, based on the gray level distributions of the background pixels. The computational cost of pixel based approaches are more acceptable than region-based methods because these methods compute the values for only one pixel at each time. However, they are very sensitive to illumination changes and image noise and cannot discriminate small changes in gray level, because of limited quantization levels. For robust and accurate background subtraction, region-based approaches are required There are many researches using region-based approaches. Liu *et al.* [31] proposed an illumination independent statistical change detection algorithm using circular shift moments (SCSM). However, because their noise estimation scheme is very heuristic, their detection is usually sensitive when a strongly uniform region exists in the images. Li and Leung [21] proposed a method based upon the integration of intensity and texture differences (IITD). They defined a texture difference measure using the cross-correlation and autocorrelation of gradient vectors of two frames. Combining different measures, they proposed a weighted integration method and a minimized energy integration method.

Most of the background subtraction and object segmentation algorithms follow a simple flow diagram shown in Figure 3.1. The four major steps in a background subtraction algorithm are preprocessing, background modeling, foreground detection, and shadow removal. Preprocessing consists of a collection of simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps. Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Foreground detection then identifies pixels in the video frame that cannot be adequately explained by the background model,

and outputs them as a binary candidate foreground mask. Finally, data validation examines the candidate mask, eliminates those pixels that do not correspond to actual moving objects, and outputs the final foreground mask. Domain knowledge and computationally-intensive vision algorithms are often used in data validation. Real-time processing is still feasible as these sophisticated algorithms are applied only on the small number of candidate foreground pixels. Many different approaches have been proposed for each of the four processing steps. I review some of the representative ones in the following subsections.

3.2.1. Preprocessing

In almost all computer vision systems, simple temporal and/or spatial smoothing is used in the early stage of processing to reduce camera noise. Smoothing can also be used to remove transient environmental noise such as rain and snow captured in outdoor camera. For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. If the camera is moving or multiple cameras are used at different locations, image registration between successive frames or among different cameras is needed before background modeling. Another key issue in preprocessing is the data format used by the particular background subtraction algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel. However, color image, in either RGB or HSV color space, is becoming more popular in the background subtraction literature. These papers argue that color is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects. In addition to color, pixel-based image features such as spatial and temporal derivatives are sometimes used to incorporate edges and motion information. For example, intensity values and spatial derivatives can be combined to form a single state space for background tracking with the Kalman filter. Pless *et al.* combine both spatial and temporal derivatives to form a constant velocity background model for detecting speeding vehicles [27]. The main drawback of adding color or derived features in background modeling is the extra complexity

for model parameter estimation. The increase in complexity is often significant as most background modeling techniques maintain an independent model for each pixel.

3.2.2. Background Modeling

Background modeling is the most important step in any background subtraction algorithm. Quite a lot research has been devoted to develop a background model that is robust against environmental and temporary changes in the background, but sensitive enough to identify all non-stationary objects of interest. I focus only on simple yet highly-adaptive techniques, and exclude those that require significant resource for initialization. The approach suggested by Jacques et al. [6] has been modified and used for background modeling. The modeling paradigm uses a model of background variation that is a bimodal distribution. The order statistics of background values during a small duration of training the system are collected and builds the framework for obtaining robust background model in the presence of undesired moving foreground objects in the field of view, such as walking people, swirling trees, etc. It is a two stage method focusing on excluding moving object pixels from the background model computation. In the first step, a pixel wise median filter is applied to several frames of the video captured for background modeling to distinguish moving pixels from stationary pixels. In the second step, only stationary pixels are processed to construct the initial background model. Let V be an array containing N consecutive images, $V^k(i,j)$ be the intensity of a pixel (i, j) in the k-th image of V, $\sigma(i,j)$ and $\lambda(i,j)$ be the standard deviation and median value of intensities at pixel (i,j) in all images in V, respectively. The initial background model for a pixel (i,j) is formed by a three-dimensional vector: the minimum $m(i,j)$ and maximum $n(i,j)$ intensity values and the maximum intensity difference $d(i,j)$ between consecutive frames observed during this training period. The background model $B(i,j) = [m(i,j), n(i,j), d(i,j)]$, is obtained as follows:

$$(1) \qquad \begin{bmatrix} m(i,j) \\ n(i,j) \\ d(i,j) \end{bmatrix} = \begin{bmatrix} minV^z(i,j) \\ maxV^z(i,j) \\ max|V^z(i,j) - V^{z-1}(i,j)| \end{bmatrix}$$

where z are the video frames which satisfy $|V^z(i,j) - \lambda(i,j)| \leq 2\sigma(i,j)$.

Haritaoglu *et al.* [13] claimed the above condition guarantees that only stationary pixels are computed in the background model, i.e., $V^z(i,j)$ is classified as a stationary pixel. After the initial training period, an initial background model $B(i,j)$ is obtained. Then, each input image $I_t(i,j)$ of the video sequence is compared to $B(i,j)$, and a pixel (i,j) is classified as a background pixel if:

$I_t(i,j) - m(i,j) \leq k\mu$ or $I_t(i,j) - n(i,j) \leq k\mu$

where μ is the median of the largest interframe absolute difference image $d(i,j)$, and k is a fixed parameter (the authors have suggested the value k equals to 2). To be more specific, if a certain pixel (i,j) has an intensity $m(i,j) < I_t(i,j) < n(i,j)$ at a certain frame t, it should be treated as background (because the value lies between the minimum and maximum values of the background model). However, the above equation may wrongly classify such pixel as foreground, depending on k, μ, $m(i,j)$ and $n(i,j)$. For example, if $\mu = 5$, $k = 2$, $m(i,j) = 40$, $n(i,j) = 65$ and $I_t(i,j) = 52$, The above condition would classify $I_t(i,j)$ as foreground, even though it lies between $m(i,j)$ and $n(i,j)$. An alternative test for foreground detection has been proposed to solve this problem, and $I_t(i,j)$ is classified as a foreground pixel if:

$I_t(i,j) > (m(i,j).k\mu) and I_t(i,j) < (n(i,j) + k\mu)$ (3)

Figure 3.2 illustrates an example of background subtraction (using $k = 2$, as in all other examples in this paper). The background image (median of frames across time) is shown in Figure 2(a), a certain frame of the video sequence is shown in Figure 2(b), and detected foreground objects are shown in Figure 2(c). It can be noticed that shadow was caused by obstruction of direct sunlight.

3.2.3. Foreground Detection

Foreground detection compares the current video frame or the image under analysis with the modeled background image, and identifies potential candidate foreground pixels from the input image. Except for the non-parametric model and the Mixed of Gaussian models, all most all the popular techniques use a single image as their background models. The most commonly used method for foreground object detection is to find out whether the input pixel is way different from the corresponding background estimate:

$|I_t(x, y) - B_t(x, y)| > T$

Another popular foreground detection scheme is to threshold based on the normalized statistics:

$\frac{|I_t(x,y)-B_t(x,y)-\mu_d|}{\sigma_d} > T_s$

where μ_d and σ_d are the mean and the standard deviation of $I_t(x, y) - B_t(x, y)$ for all spatial locations (x, y). Most schemes determine the foreground threshold T or Ts experimentally. Actually, the threshold should be a function of the spatial location (x, y). For example, the value of the threshold should be smaller for regions with low contrast. Another method to have spatial variability is to use hysteresis based two thresholds.

Even though foreground object detection and segmentation sounds like a simple problem, it often generates small false-positive or false-negative regions. Generally, non-stationary pixels from moving trees and leaves or shadow cast due to blocking of light by moving objects are often mistaken as true foreground objects. In order to eliminate the false-positive objects resulting from moving trees, the background model must adapt to the changes and take care of this problem. The undesired objects resulting from shadow casts are eliminated by using the shadow detection and removal algorithm explained in the next section.

3.2.4. Shadow Detection and Removal

Shadows appear in an image when objects totally or partially block direct light from a source of illumination. According to the classification described in [15], shadows are classified into two classes: cast and self shadows. A cast shadow is projected by the object present in the scene in the direction of the light source. On the other hand, a self shadow is the part of the object of interest which is not properly illuminated by direct light. The presence of cast shadows in an image modifies the perceived object shape. Whereas, the presence of self shadows modify the perceived object shape and its color. In order to represent the object shapes correctly, shadows must be identified and removed. However, neither moving object segmentation nor change detection techniques can discriminate between moving objects and moving shadows. Moving shadows cause the undesired segmentation of objects in the scene and changes the overall shape of the object. An example of improper segmentation and its correction by means of shadow removal is shown in Figure 3.2.

According to [12], it is expected that a certain fraction of incoming light is blocked to create a shadowed region. Even though there are several different factors which influence the intensity of a pixel in a cast shadow region [4], I found that the observed intensity of shadow casted pixels is directly proportional to incident light. As a result, the intensity of shadowed pixels are scaled versions (mostly darker) of corresponding pixels in the reference background model. As observed by authors in [7], the normalized cross correlation (NCC) is a promising method to detect shadow pixel candidates. Since NCC is often used to detect the scaled versions of the same signal. In this research, I used the NCC as the first step for shadow detection, and refine the result using local statistics of pixel ratios, as explained below.

Suppose $B(i, j)$ be the modeled background reference image formed by adaptive background modeling discussed earlier, and $I(i, j)$ be an image of the captured video sequence or a static two dimensional image. For each foreground pixel (i, j), consider

(a) Background Model

(b) Current Video Frame

(c) Segmented Foreground Object(The boundary is outlined by a red color rectangle)

(d) Segmented Binarized Object with Shadow

(e) Segmented Binarized Object after Shadow Removal

FIGURE 3.2. Results of Background Subtraction and Video Object Segmentation

a $(2N+1) \times (2N+1)$ template T_{ij} such that $T_{ij}(n,m) = I(i+n, j+m)$, for -$N \leq n \leq N$, -$N \leq m \leq N$ (i.e. T_{ij} represents to a neighborhood of pixel (i,j)). The NCC between template T_{ij} and image B at pixel (i,j) is given by the following equation:

$$NCC(i,j) = \frac{ER(i,j)}{E_{B(i,j)} E_{T_{ij}}}, \tag{2}$$

where

$$ER(i,j) = \sum_{n=-N}^{N} \sum_{m=-N}^{N} B(i+n, j+m) T_{ij}(n,m), \tag{3}$$

$$E_{B(i,j)} = \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} B(i+n, j+m)^2}, \tag{4}$$

$$E_{T_{ij}} = \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} T_{ij}(n,m)^2}, \tag{5}$$

For a pixel (i, j) in a shadowed region, the NCC in a neighboring region T_{ij} should be large (close to one), and the energy $E_{T_{ij}}$ of this region should be lower than the energy $E_{B(i,j)}$ of the corresponding region in the background image. Thus, a pixel (i, j) is pre-classified as shadow if:

$$NCC(i,j) \geq L_{ncc} \text{ and } E_{T_{ij}} < E_{B(i,j)}, \tag{6}$$

where L_{ncc} is a fixed threshold. If L_{ncc} is low, several foreground pixels corresponding to moving objects may be misclassified as shadows. On the other hand, selecting a larger value for L_{ncc} results in less false positives, but pixels related to actual shadows may not be detected. In fact, the influence of the threshold L_{ncc} for shadow detection has been observed for different scenarios and an optimal value has been chosen through experiments providing best results for all sorts of background scenes. Experiments with different backgrounds brought a conclusion that choosing $N = 5$ and $L_{ncc} = 0.95$ results in a reasonable balance between false positives and false negatives.

3.2.5. Shadow Refinement

The NCC provides a good initial estimate about the location of shadowed pixels, by detecting pixels for which the surrounding neighborhood is approximately scaled with respect to the reference background. However, some background pixels related to valid moving objects may be wrongly classified as shadow pixels. To remove such false positives, a refinement stage is applied to all pixels that satisfy Equation 6. The proposed refinement stage consists of verifying if the ratio $I(i,j)/B(i,j)$ in a neighborhood around each shadow pixel candidate is approximately constant, by computing the standard deviation of $I(i,j)/B(i,j)$ within this neighborhood. More specifically, I consider a region R with $(2M+1) \times (2M+1)$ pixels (I used M = 1 in all experiments) centered at each shadow pixel candidate (i,j), and classify it as a shadow pixel if:

$$std_R\left(\frac{I(i,j)}{B(i,j)}\right) < L_{std} \text{ and } L_{low} \leq \left(\frac{I(i,j)}{B(i,j)}\right) < 1, \tag{7}$$

where $std_R\left(\frac{I(i,j)}{B(i,j)}\right)$ is the standard deviation of quantities $I(i,j)/B(i,j)$ over the region R, and L_{std},L_{low} are thresholds. More precisely, L_{std} controls the maximum deviation within the neighborhood being analyzed, and L_{low} prevents the mis-classification of dark objects with very low pixel intensities as shadowed pixels. The values of L_{std} = 0.05 and L_{low} = 0.5 are obtained experimentally. Some morphological bridging operators are applied to fill the holes and filtering is done to remove isolated pixels.

3.3. Object Segmentation

Reliable object segmentation plays an important role for object recognition and classification. Improper segmentation of the object from the image or the input video frame adversely affects the performance of of object tracking system. Careful measures are to be taken while segmenting the objects on interesting while ignoring the noise and undesired components. Blob analysis has been proved to be a reliable tool to provide information about the objects to the segmentation algorithm. The

segmentation module carefully analyzes the features of all the objects present in the scene and filters the objects who satisfy certain criteria.

3.3.1. Blob Analysis

In image processing, a blob is defined as a region of connected pixels. Blob analysis is the identification and analysis of these connected pixel regions in an image. The algorithms distinguish pixels by their value and mark them in one of two categories: the foreground (typically pixels with a non-zero value) or the background (pixels with a zero value).

Most applications that use blob analysis, the blob features frequently calculated are area and perimeter, Feret diameter, blob shape, number of holes, and location. The skillfulness of blob analysis tools makes them suitable for a wide variety of applications.

Since a blob is a region of connected pixels, analysis tools generally consider touching foreground pixels to be part of the same blob. As a result, what is easily identifiable by the human eye as several distinct but touching blobs may be interpreted as a single blob by the algorithm. Furthermore, any part of a blob that is in the background pixel state because of lighting or reflection is considered as background during analysis.

3.3.2. Blob Filter

Since noise components are frequently present in an image, it is important to ignore the undesired objects from the image after background subtraction and thresholding. Sometimes, improper selection of the threshold generate small sized foreground objects considered for segmentation. So, a careful design of the filter is highly needed to segment valid objects only. The filter eliminates the small sized noise blobs based on the area. Similarly, other blob features like the aspect ratio and compactness are used to filter the undesired blob components.

3.4. Summary

This chapter discussed the process used for background subtraction and object segmentation. In practice, some morphological opening and closing operations are applied intermediately to retain the shape of the object.

CHAPTER 4

FEATURE EXTRACTION AND CLASSIFICATION

4.1. Introduction

Feature extraction holds an important stepping stone to pattern recognition and machine learning problems. To recognize and classify an object in an image, I must first extract some features out of the image. Feature extraction is the technique to extract various image attributes for identifying or interpreting meaningful physical objects from images. The primary idea is to represent the visual appearance of an object by distinctive key features, or attributes. Once the object is segmented, carefully chosen features are extracted to perform the desired recognition task using this reduced representation instead of the full size object image. It is often decomposed into feature construction and feature selection. Now under different conditions (e.g. lighting, background, changes in orientation etc.) the feature extraction process will find some of these distinctive keys, but in general not all of them. However, the fraction that can be found by existing feature extraction processes is frequently sufficient to identify objects in the scene. This addresses one of the principle problems of object recognition, which is that, in any but rather artificial conditions, it has so far proved impossible to reliably segment whole objects on a bottom-up basis. In the current system, local features based on automatically extracted boundary fragments are used to represent views (aspects) of rigid 3-D objects, but the basic idea could be applied to other features and other representations. The objective is to identify the most discriminative image features with the lowest dimensionality in order to reduce the computational complexity and improve the tracking accuracy. In traditional pattern recognition, linear discriminant analysis (LDA) and principal component analysis (PCA) are widely used to reduce the dimensionality. The different features I used in

my experiments are invariant moments, traditional curve matching, fourier descriptors of curve Segments, B-spline coefficients. Before I computed the features of the object shape, the whole object was represented by statistical morphological skeleton. So, the skeleton extraction was the very first step in feature extraction.

4.2. Statistical Skeleton Extraction

Recently, there is an increasing number of studies on feature extraction and selection reveals that it is difficult to select features accomplishing accuracy of object localization, consistency of detection, and having low computational complexity. A highly information-preserving shape-descriptor called as morphological skeleton (MS) [33], shows all these characteristics. However, this description is noise dependent as described by authors in [25]. To eliminate this dependency, a new noise tolerant 2-D shape descriptor, i.e., the statistical morphological skeleton (SMS) [10], has been considered as a common feature for object shape representation. Statistical skeleton extraction is obtained by using an algorithm based on a new class of parametric binary morphological operators, considering statistical aspects. Parameters are adaptively selected during the successive iterations of the skeletonization operation to regulate the properties of the object shape descriptor. Shape representation results show the greater robustness of the proposed method as compared with other morphological approaches. Finally, object recognition is obtained by comparing an analytical approximation of the skeleton function extracted from the analyzed image with that obtained from model objects stored into a database. Tracking is performed by computing a set of observable quantities derived from the detected SMS and other geometric characteristics of the moving object.

The basic technique for morphological skeleton extraction has been described by Maragos *et al.* [25]. Erosion ($I \ominus S$) is an operation in which a structuring element of a shape (say, 3x3 array of 1's) is placed at every position of the binary image, the pixel value at that position in the resultant image is set to minimal value obtained by logical AND operations on the corresponding pixels of I (segmented binary Image)

and S (structuring matrix). Similarly, dilation is obtained by considering maximum value(denoted by $I + S$). Statistical skeleton can be extracted by considering the noise level. I compute at each pixel position m

$E_m = \frac{IS_m \times exp(-\beta)}{(N + IS_m(exp(-\beta) - 1))}$

where b is a parameter chosen based on image noise level, N is the number of elements in the structuring element S, and IS_m is the number of pixels of I overlapping S placed at m. H_m is computed similarly with β replacing -β above.

$H_m = \frac{IS_m \times exp(\beta)}{(N + IS_m(exp(\beta) - 1))}$ By choosing the image pixel value at each m as 1 based on $E_m > \theta$ (or $H_m > \theta$) or not, where θ is a pre-chosen threshold, I get statistical equivalent of erosion (or dilation) operation.

The algorithm, say SSE(I), considers an image I as input and obtains as output an image representation $R(I)$ corresponding to the statistical skeleton. In the following iterative algorithm, I use values of β (β_is) that increase with iteration i. The statistical erosion and dilation operators are subscripted by β_is.

(1) Initialization: $i = 1$, $I_0 = I$, the given binary image

(2) $I_i = (I_{i-1} S_{\beta i})$; if $I_i = [0]$, $R_i = I_{i-1}$; stop;

(3) $R_i = I_{i-1} \ \ (I_i + S_{\beta i}$); (- differs to set difference operation)

(4) $i = i + 1$; $\beta i = ln(gain * i) + offset$; if $i > i_{max}$: stop; else go to step 2;

Note: i_max, $gain$ and $offset$ are chosen parameters to make it possible to apply linear filters at the initial steps and morphological operators at the final stages.

The skeleton is formed by the combination of representations at intermediate steps R_is obtained in the successive iterations.

i.e.

$$SSE(I) = \cup R_i(I) \tag{8}$$

Statistical opening consists of the superposition of two effects: noise filtering, mainly due to binary statistical erosion (BSE) performed at the first step, and shape approximation obtained by successively applying binary statistical dilation (BSD) to

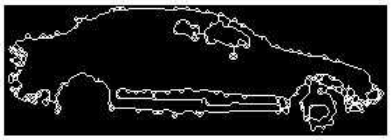
the result of BSE. The noise-filtering effect progressively disappears when $\beta \to \infty$, while shape approximation becomes coarser when β increases. The reason for using a logarithmic scheduling approach to selection of β values can now be interpreted in a more complete way. At the first steps, when a low β value is used not only is noise eliminated (BSE effect), but also boundaries of the shape are better approximated (BSD effect). In fact, for $\beta = 0$ the statistical opening corresponds to twice the application of a linear average thresholded at θ. This double averaging operation allows one to smooth the object boundaries in a way that also depends on the selected structuring element. The smoothing effect is progressively relaxed when β increases, as noise is supposed to disappear and boundaries to converge to a shape which fits well to the chosen structuring element. The logarithmic scheduling allows convergence to the smoothed boundary to occur in a continuous fashion.

4.3. Feature Extraction

4.3.1. Geometric Moments

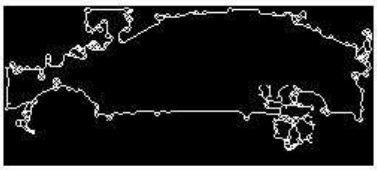
The mathematical concept of moments has been around for many years and has been utilized in many fields ranging from mechanics and statistics to pattern recognition and image understanding. Describing images with moments instead of other more commonly used image features, means that global properties of the image are used rather than local properties. Geometric moment invariant was first introduced by Hu [14]. The definition was derived from the theory of algebraic invariant. From methods of algebraic invariants, he derived a set of seven moment invariants, using non-linear combinations of geometric moments. These invariants remain the same under image translation, rotation and scaling. Since then, moments and functions of moments are widely used in pattern recognition [2], ship identification [3], aircraft identification [4], pattern matching [5] and scene matching [6].

(a) Input Video Frame



(b) Statistical Morphological Skeleton

(c) Input Video Frame



(d) Statistical Morphological Skeleton

FIGURE 4.1. Results of Statistical Morphological Skeleton

A general definition of moment functions Φ_{pq} of order $(p+ q)$, of an image intensity function $f(x, y)$ can be defined as follows:

$$\Phi_{pq} = \int_x \int_y \Psi_p q(x, y) f(x, y) dx dy, \tag{9}$$

where $\Psi_p q(x, y)$ is the moment weighting kernel. The basis functions may have a range of useful properties that may be passed onto the moments, producing descriptions which can be invariant under rotation, scale and translation. Geometric moments are the most popular types of moments and are frequently used for a number of image processing tasks. The two-dimensional geometric moment of order $(p + q)$ of a function $f(x, y)$ is defined as

$$m_{pq} = \int_{\infty}^{\infty} \int_{\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{10}$$

The two dimensional geometric moment for a $(N \times N)$ digital image is given by

$$m_{pq} = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} x^p y^q f(x, y) \tag{11}$$

The monomial product $x^p y^q$ is the basis function for the moment definition described above. So, geometric moments are not orthogonal since the basis function itself is not orthogonal. However, the uniqueness theorem states that the moment set m_{pq} is unique for a given image $f(x, y)$.

For the segmented binary objects I have:

$$m_{pq} = \sum_{I} x^p y^q \tag{12}$$

where the summation runs over all the elements in I, i.e., all the foreground pixels in the image matrix.

With this framework I compute shape features or measurements which are invariant to certain affine transformations.

The moments of $f(x, y)$ translated by an amount (a,b), are defined as,

$$m_{pq} = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} (x+a)^p (y+b)^q f(x,y) \tag{13}$$

Thus, the central moments can be computed by replacing a with -x_c and b with -y_c.

$x_c = \frac{m_{10}}{m_{00}}$ and $y_c = \frac{m_{01}}{m_{00}}$

So, the central moments on the binary image can be computed by the following equation:

$$m_{pq} = \sum_{I} (x - x_c)^p (y - y_c)^q \tag{14}$$

where the summation runs over all foreground elements in I.

When a scaling normalization is applied the central moments, the moments become scale invariant. The equation to do the same is described below

$$\eta_{pq} = \frac{m_{pq}}{m_{00}^{\gamma}}, \quad \gamma = \left[\frac{p+q}{2}\right] + 1 \tag{15}$$

Hu has defined seven moment values, using normalized central moments upto order three. These moment values are invariant to object position, orientation and scale. The formulas for seven moments are given below:

$$
\begin{aligned}
M_1 &= \eta_{20} + \eta_{02} \quad M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
M_6 &= (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\quad +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
M_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad +(3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned}
\tag{16}
$$

4.3.2. Curve Matching

Quite a few algorithms have been developed for the two-dimensional object recognition and location problem. I mention a few of them, which use curve matching. Freeman [11] describes recognition of two dimensional shapes by sets of well defined critical points (such as discontinuities in curvature) and computes shape features between consecutive critical points locally. This method, however, does not work for curves which do not possess such critical points, or in object images where such points are occluded. Ayache and Faugeras [1] discussed object matching on finding correspondence between sides of polygons, which approximate the original curves. This requires polygonal approximations to be reasonably stable in the number and relative length of corresponding sides in the resulting polygons. This method is general and does not require a polygonal approximation of the curves. (I do use such an approximation to smooth the curves, but it is not essential to the matching technique.) The problem which is tackled in [1] is, however, more complex, since they allow translation, rotation, and scale change of the observed objects. Even though I do solve for the scale change, I use a different technique the estimate the scale change and resize the object to a standard size. I plan to extend my recognition method to a more

robust scale invariant case in future. Turney *et al.* [16] demonstrated recognition is obtained by creating sub templates of the models and matching them against the boundary of the scene. Kalvin *et al.* [17] introduced an object recognition technique which is particularly efficient when a large database of models is involved. The deficiency of this technique is the need to use the so called breakpoints, which were mentioned before. An improvement of this algorithm which eliminates the need of breakpoints has been recently developed.

4.3.3. Curve Matching Approach

In many object recognition and content-based image retrieval applications, the object shapes are represented by their boundary curves. And Curves are matched by using some curve matching algorithms for recognition and classification. Boundary curves typically do not represent interior details of the objects. Despite this well-known disadvantage, it has been extensively used in many computer vision applications [8, 24]. Matching typically involves finding a mapping from one curve to the other. The two-dimensional boundary curve matching algorithm I implemented for object recognition draws largely from algorithms by Schwartz & Sharir [32] and Wolfson [35]. I found these algorithms suitable for object recognition because they have been proved promising for arbitrary curves. Moreover, the partial matching component of Wolfsons algorithm shows promise for reliable matching and recognition of curves derived from noisy statistical morphological skeletons of objects where occlusions and illumination changes can easily cause fragmented curves, in comparison with the ideal curves generated from clearly visible objects.

To enhance the mentioned algorithms for object recognition under the constraints, I added a curve connection option and a technique for interpolating matches after the best partial matches have been determined. I also introduced several parameters for the algorithms so that knowledge of constraints on matching can be used to control allowable matches. There are four steps in the curve matching scheme:

- Curve Smoothing: The boundary curve is extracted from the statistical morphological skeleton of the object. The curve is smoothed by the smoothing algorithm suggested by Schwartz & Sharir [32]. This method computes the shortest path within an epsilon neighborhood of the curve. I use heuristic based curve smoothing because the different transformations the object undergo because the extracted and expected two-dimensional curve sets do not necessarily satisfy the conditions of a lemma [32]. This is due to the change in illuminations and varying distance from the camera. So, it justifies the smoothing operation for other matching problems.
- Finding Partial Matches: In this step a list of reasonable matches, including partial matches, is constructed. I have adapted the shape signature string matching algorithm explained by Wolfson [35] which achieves partial matching by comparing approximations of curvature as a function of curve length. The output of this algorithm generates a large number of pairs of partially matching curves and sub-curves. The objective is to quickly build a list of large number of promising matches, discarding those which are obviously wrong. I discard curves based simple heuristic parameters.
- Selecting Best Matches: The objective of this step is to find the best match of a curve segment against the curve segments of all the objects stored in the database. This is used primarily to register the object on scene by finding out the orientation of the curve segment with respect to the original object boundaries. I use a process of elimination, rejecting matches that do not satisfy the matching score which is defined as a parameter. The matching score is the output of a cross-correlation function between the curve under inspection and stored curve segments from objects during registration process explained in introduction chapter.
- Final Matching: To complete the curve matching method for object recognition and classification, the set of final partial curve segment matches is

examined to find pairs of more number of curves between which two or more curve segments have been partially matched. In this case, the curve segments are taken from the same object. A new matching score is computed while combining the matching scores of individual curve segments. This post processing of the curve matching process is intended to take care of the partially occluded objects. These are natural consequences of the presence of trees, buildings and other vehicles on the scene. A lot of researchers have suggested many additional strongly model-based ways of proceeding from this point that I have not considered.

4.3.4. B-splines

Even though, moments have been used extensively for object boundary curve recognition and classification in many computer vision applications moments are noise sensitive, are unreliable under random sampling, and cannot estimate affine transform parameters accurately. B-spline is one of the most effective curve representations as described by many researchers. A discrete curve can be modeled by a continuous one by using B-spline curve fitting. Only a small set of B-spline coefficients are needed for curve representation and matching. Additionally, B-spline representation is complete, compact, and robust to noise and random sampling of the curve points.

The object skeleton extracted from the step statistical morphological skeleton extraction is divided in to a number of cells based on the division parameter. The object shape curves present in one cell are smoothed and interpolated using B-spline curve fitting. The B-spline coefficients (control points) are then estimated. They are used as object shape features and kept in the database during the registration of a new object or vehicle intended to be tracked. When the object or vehicles in a video frame or static image are to be recognized and tracked, the system tries to match the query shape with the registered shapes. The query shape must be smoothed and interpolated before estimating B-spline coefficients. The knot points, which are one to one transformation of the control points, from the query control points and the

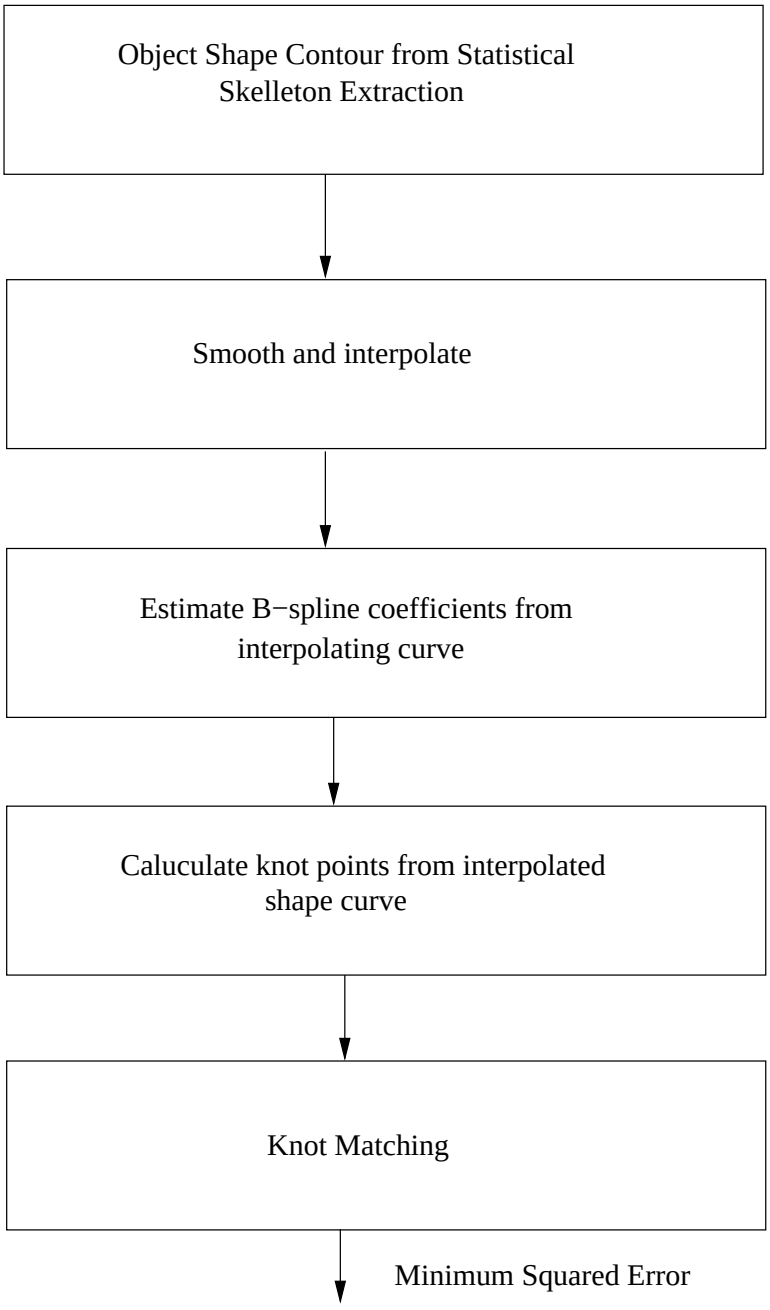


FIGURE 4.2. Block Diagram of B-spline Based Object Shape Recognition

registered control points are estimated and matched using B-spline curve matching technique. Finally, a k-nearest neighbor classifier is used to classify the query shape curve into one of the tracked object numbers. This process is continued for all the cells and finally the object under inspection is either classified into one of the known tracked object names or rejected.

Curve Modeling

Assume that I are given a dense set of m data curve points si, j = O,..,m- 1. The primary goal is to model the input curve using closed cubic B-splines that consist of n+l connected curve segments r_i, i = 0,1,..,n. Each of these segments is a linear combination of four cubic polynomials in the parameter $t \in [0\ 1]$

$$r_i(t) = C_{i-l}Q_0(t) + C_iQ_1(t) + C_{i+l}Q_2(t) + C_{i+2}Q_3(t) \tag{17}$$

for i = 0,1,..,n, where $Q_k(t) = a_{k0}t^3 + a_{k1}t^2 + a_{k2}t^1 + a_{k3}t^0$, k = 0,1,2,3.

Using the continuity constraints in position, slope and curvature on the connection points between segments and the invariance property to coordinate transformations $\sum_{k=0}^{3} Q_k(t) = 1, t \in [0\ 1]$ the polynomial factors a_k are computed and thus the basis functions $Q_k(t)$ are defined. The B-spline used to model the input curve is given using the curve segments as:

$$r(t') = \sum_{k=0}^{n} r_i(t'-i) = \sum_{k=0}^{n} C_{imod(n+l)}N_i(t') \tag{18}$$

where $0 \le t' < n-2$ and $N_i(t')$ denote the so-called blending functions:

$$N_i(t') = \begin{cases} Q_3(t'-i+3) & i-3 \le t' < i-2 \\ Q_2(t'-i+2) & i-2 \le t' < i-1 \\ Q_1(t'-i+1) & i-1 \le t' < i \\ Q_0(t'-i) & i \le t' < i+1 \\ 0 & otherwise \end{cases} \tag{19}$$

In order to find the appropriate B-spline, the control points C_i must be determined. The approach followed in this work tries to find an approximate B-spline such that the error between the observed data and their corresponding B-spline curve is minimized. In this sense, the metric $d^2 = \sum_{j=1}^{m} \left\| s_j - r(t'_j) \right\|^2$ should be minimized. If appropriate parametric values of t' are allocated on the curve, then the minimum

mean square error (MMSE) solution for the control points is given in matrix form as $C^f = (P^T P)^{-1} P^T f$ where f and C_f are of size $m \times 2$ and $(n+1) \times 2$ are respectively containing the given data points s_j and the control points $C, \_i$ respectively. The $m \times (n+l)$ matrix P contains appropriate values for the blending functions, estimated on the points $r(t'_j)$, as shown in the following equation.

(20)

$$P = \begin{pmatrix} N_0(t'_1) + N_{n+1}(t'_1) & N_1(t'_1) + N_{n+2}(t'_1) & N_2(t'_1) + N_{n+3}(t'_1) & \cdots & N_3(t'_1) & N_n(t'_1) \\ N_0(t'_2) + N_{n+1}(t'_2) & N_1(t'_2) + N_{n+2}(t'_2) & N_2(t'_2) + N_{n+3}(t'_2) & \cdots & N_3(t'_2) & N_n(t'_2) \\ \vdots & & \vdots & & & \vdots \\ N_0(t'_m) + N_{n+1}(t'_m) & N_1(t'_m) + N_{n+2}(t'_m) & N_2(t'_m) + N_{n+3}(t'_m) & \cdots & N_3(t'_m) & N_n(t'_m) \end{pmatrix}$$

4.3.5. B-spline Curve Matching

Representing the shapes using B-spline knot points can reduce the dimension of feature vector from the original data points to just small number of knot points. However, before matching the knot points between the sample shapes in the database and the test shape, the estimation of the affine transform should be performed to correct any transformation present. Since an affine transformation on a B-spline generates a B-spline, its control points are obtained by transforming the original control points with those affine-transform parameters. Because of the non-uniqueness of the control points in representing curves, I can not directly use the estimated control points in the matching process.I use a similarity measure between two curves based on their Bspline knot points for matching.

4.4. Classification

A commonly used classification method in pattern recognition called as nearest neighbor (NN) classification is used primarily for classification. First, I create a database containing the desired shape properties of sample objects and provide a distinguishable class name to a distinct object. During recognition, when the system

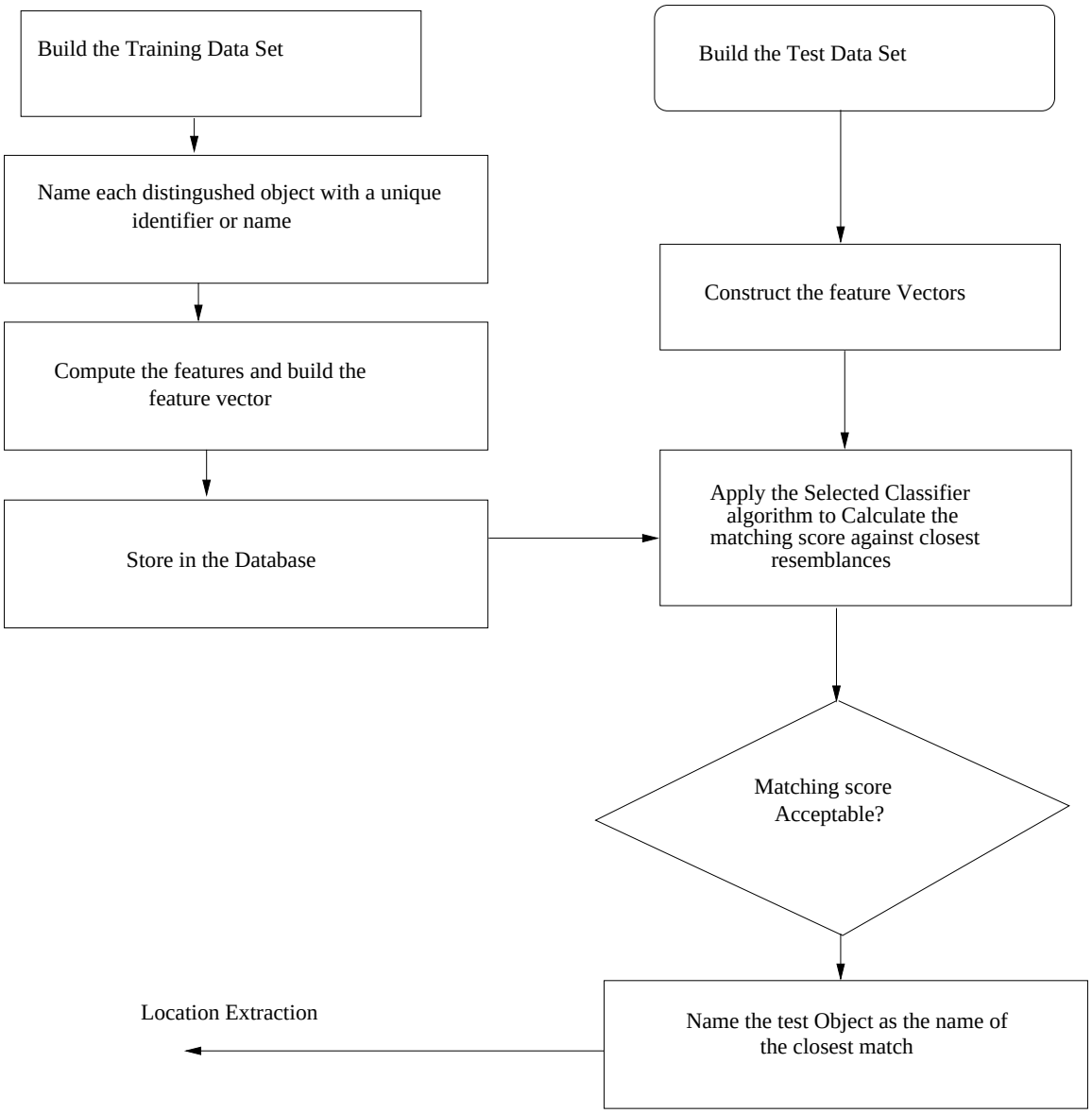


FIGURE 4.3. Architecture of Recognition and Classification Module

is given a query shape (mostly the representing features), i.e., a test object shape to classify, the system simply finds the nearest neighbor of the query in the database. The classification scheme uses eucledian distance metric to compute the similarity measure to find the database object that is the most similar to the query. Then, the classification system classifies the query object as belonging to the same class as its nearest neighbor. For example, if the query object is an image of a vehicle, and the nearest neighbor of the query in the database is an image of the vehicle with class

number '3', then the system classifies the query object as an image of "3". Sometimes, instead of looking at the single nearest neighbor, a classification system uses the K nearest neighbors (where K can be any number) to classify the query object. The query object gets classified to that class which has majority among the K nearest neighbors. Figure 4.3 outlines the schematic architecture of the recognition process used in this research work.

4.5. Summary

This chapter discussed the different features considered to represent the object shape. The iterative process used to extract statistical morphological skeleton shows promising results to represent any object shape. Simple classifiers have been discussed to make the implementation of the system simple.

CHAPTER 5

RESULTS AND DISCUSSION

In order to test the performances of the different algorithms proposed to build the tracking system, several videos are captured and frames are extracted from the video for the training as well testing purposes. A number of different vehicle classes are considered for building the training data set after segmenting from real-time traffic scenes. Similarly, test data was generated considering several video sequences. The efficiency and performance of the system is assessed in terms of the recognition accuracy and response time.

5.1. Object Shapes and Varieties

The different types of objects I used included objects of similar shapes but different sizes as well as objects with very different shapes. I label the objects with identifiable numbers for easy reference later in the test cases and results. Some of these objects are shown in the Fig. 5.1 and Fig 5.2.

5.2. Implementation Details

All the algorithms discussed in this thesis are implemented in matlab on a computer having P-IV processor. Some of the inbuilt functions present in matlab are used to speed up the experiment.

5.3. Recognition Accuracy of Object Shapes

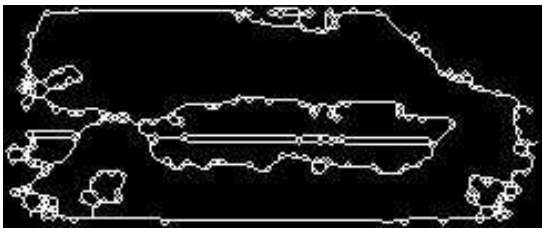
In order to calculate the recognition accuracy, two different data sets were constructed. One being the training set and other test set. Training set contains at least one sample of a vehicle shape later used for tracking. The vehicles were segmented from real-time traffic scene and the representative features were stored in the database. Each of the test set data object underwent similar processing as that of

(a) Background Scene

(b) Test Object 1

(c) Skeleton Representation of Test Object 1

(d) Test Object 2



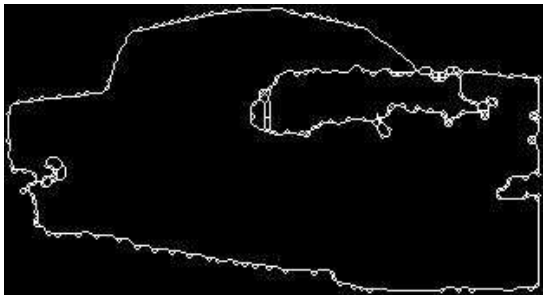
(e) Skeleton Representation of Test Object 2

FIGURE 5.1. Samples Test Data Set 1

training set. In the experiment, i have used almost 30 different classes of vehicles in the training set from noise free frames. And approximately 100 different frames were

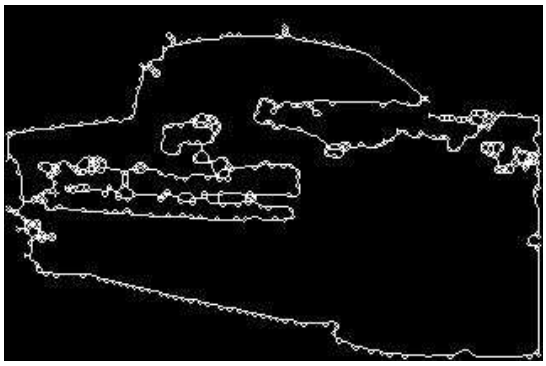
(a) Background Scene

(b) Test Object 1



(c) Skeleton Representation of Test Object 1

(d) Test Object 2



(e) Skeleton Representation of Test Object 2

(f) Test Object 3

(g) Skeleton Representation of Test Object 3

FIGURE 5.2. Samples Test Data Set 2

TABLE 5.1. Recognition Accuracy (in percent) of Test Object Shapes Using Different Features and Classifiers

Feature Type	NN Classifier	k-NN Classifier
Invariant Moments	90.5	91.2
Simple Curve Matching	84.5	86.6
B-spline Curve Matching	93.4	94.3

considered for building the test dataset. The following table outlines the recognition accuracy by different methods.

5.3.1. Recognition Using Invariant Moments

The seven invariant moments are calculated for the training set and stored in the database. The same seven affine invariant moments are also calculated for all the samples in test dataset and classified each of them using both NN and k-NN classifier. This feature demonstrated promising results for objects not having any occlusion. So, it could be used in a tree based classifier where occlusion could be detected a priori.

5.3.2. Recognition Using Simple Curve matching

This produced the lowest recognition accuracy. on the top of that the calculation of cross-correlation was not computationally inexpensive.

5.3.3. Recognition Using B-spline Coefficients

The grid based shape descriptor, described in the previous chapter, is chosen for matching using B-spline coefficients. The segmented object is resized to a standard size after estimating the affine transform parameters. The grid cells are then scanned from left to right and top to bottom. For each cell, the curves are approximated using B-spline estimation. B-spline parameters are calculated from each cell and used to build the feature vector. The feature vector constructed after calculating the parameters from each cell is matched against stored object feature vectors in the database. The database usually keeps at least one feature vector for one distinguished vehicle. This recognition and classification strategy returned higher accuracy compared to other representative features used in this thesis.

5.4. Classifier

The design of adaptive classifier plays an important role for recognition and classification of complex objects. I have considered two simple classifiers such as nearest neighbor (NN) and k-nearest neighbor (k-NN) classifiers for simplicity. The value of k has been taken as 3 during classification.

5.5. Summary

This chapter has discussed the performance of the system. The recognition accuracy of the system while using different representative features is tabulated. Since the response time of the object tracking systems reported in literature is not available, I could not compare the response time with other methods available in the literature. However, most of the important steps of this system could be done with in 1 minute for one object in the configured environment set up for the experiments.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

Automatic object recognition and tracking has the potential to improve identification and event analysis of video sequences. It is vital in video-based remote surveillance and prediction of potential threats. To achieve this goal, several techniques have been discussed and presented to detect and segment video objects. Each of the underlying steps are analyzed carefully considering the quality of the video and images. To conclude this thesis, I summarize the major contributions of this work, then outline several directions for future work to improve the performance of the system.

The designed and implemented background subtraction and object segmentation framework combines various image and video processing algorithms discussed in literature. The selection of features for better recognition accuracy are obtained mostly from experiments. This thesis has addressed a number of challenging issues associated with object segmentation when the quality of the image is poor and noisy. The challenge of recognizing partially occluded objects is addressed and required measures are taken to achieve satisfactory recognition accuracy. The algorithms have been tested on a number of video sequences captured with different cameras to prove the credentials of the proposed system.

This research work will lay a stepping stone for the further developments of the automatic vehicle tracking system in a secured area. Among such extensions, application specific modifications, dynamic and adaptive background modeling, a tree based hierarchical classifier could be considered in future work. More shape representing features could be considered for better recognition accuracy for partially occluded objects.

BIBLIOGRAPHY

[1] N Ayache and O D Faugeras, *Hyper: a new approach for the recognition and positioning to two-dimensional objects*, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986), no. 1, 44–54.

[2] E. Bala and A. E. Cetin, *Computationally efficient wavelet affine invariant functions for shape recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, 2004, pp. 1095–1099.

[3] R. H. Bartels; J. C. Beatty; and B. A. Barsky, *An introduction to splines for use in computer graphics and geometric modeling*, Morgan Kaufmann Publishers, 1987.

[4] E. Salvador; A. Cavallaro; and T. Ebrahimi, *Cast shadow segmentation using invariant color featurese*, Computer Vision and Image Understanding, vol. 95, August 2004, pp. 238–259.

[5] Shih-Fu Chang, William Chen, Horace J. Meng, Hari Sundaram, and Di Zhong, *Videoq: An automated content based video search system using visual cues*, ACM Multimedia, 1997, pp. 313–324.

[6] Jacques J.C.S.; Jung C.R.; and Musse S.R., *Background subtraction and shadow detection in grayscale video sequences*, 18th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), October 2005, pp. 189–196.

[7] D. Grest; J.-M. Frahm; and R. Koch, *A color similarity measure for robust shadow removal in real time*, In Vision, Modeling and Visualization, 2003, pp. 253–260.

[8] Yoram Gdalyahu and Daphna Weinshall, *Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes*, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999), no. 12, 1312–1328.

[9] A. Diplaros; T. Gevers; and I. Patras, *Combining color and shape information for illumination-viewpoint invariant object recognition*, IEEE Transactions on Image Processing, vol. 15, 2006, pp. 1–11.

[10] Foresti G.L. and Regazzoni C.S., *Statistical morphological skeleton for representing and coding noisy shapes*, IEE Proceedings on Vision, Image and Signal Processing, vol. 146, August 1999, pp. 85–92.

[11] Freeman H., *Shape description via the use of critical points*, PR 10 (1978), no. 3, 159–166.

[12] A. Elgammal; R. Duraiswami D. Harwood; and L. Davis, *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance*, Proceeedings of the IEEE, vol. 90, 2002, pp. 1151–1163.

[13] I. Haritaoglu; D. Harwood; and L. Davis, *Realtime surveillance of people and their activities*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, August 2000, pp. 809–830.

[14] Ming-Kuei Hu, *Visual pattern recognition by moment invariants*, IEEE Transactions on Information Theory, vol. 8, August 1962, pp. 179–187.

[15] C. Jiang; and M.O. Ward, *Shadow identification*, Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition, 1992, pp. 606–612.

[16] Turney J.L.;, Mudge T.N., and Volz R.A., *Recognizing partially occluded parts*, PAMI 7 (1985), no. 4, 410–421.

[17] Alan Kalvin, Edith Schonberg, Jacob T. Schwartz, and Micha Sharir, *Two-dimensional model-based boundary matching using footprints*, Int. J. Rob. Res. 5 (1987), no. 4, 38–55.

[18] N. Katzir, M. Lindenbaum, and M. Porat, *Curve segmentation under partial occlusion*, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994), no. 5, 513–519.

[19] M. Hahnel; D. Klunder; and K. Kraiss, *Color and texture features for person recognition*, IEEE International Joint Conference on Neural Networks, vol. 1, July 2004, pp. 647–652.

[20] D. Koller, *Moving object recognition and classification based on recursive shape parameter estimation*, 1993.

[21] Li L. and Leung M.K.H., *Integrating intensity and texture differences for robust change detection*, IP 11 (2002), no. 2, 105–112.

[22] S. Belongie; J. Malik; and J. Puzicha, *Shape matching and object recognition using shape contexts*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, 2002, pp. 509–522.

[23] Gerard Medioni and Yoshio Yasumoto, *Corner detection and curve representation using cubic b-spline*, Comput. Vision Graph. Image Process. 39 (1987), no. 3, 267–278.

[24] E. Milios and E. Petrakis, *Shape retrieval based on dynamic programming*, 2000.

[25] Maragos P. and Schafer R., *Morphological skeleton representation and coding of binary images*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34, October 1986, pp. 1228–1244.

[26] S. Pansang and C. Kimpan, *3-d object recognition from shadow shape curves*, The IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS 2000), 2000, pp. 437–440.

[27] Pless R., Larson J., Siebers S., and Westover B., *Evaluation of local models of dynamic backgrounds*, CVPR03, 2003, pp. II: 73–78.

[28] Lourena Rocha, Luiz Velho, and Paulo Cezar Pinto Carvalho, *Image moments-based structuring and tracking of objects*, SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing (Washington, DC, USA), IEEE Computer Society, 2002, pp. 99–105.

[29] P. Rosin, *Thresholding for change detection*, ICCV'98, 1998, pp. 274–279.

[30] Cheung S.-C. and C. Kamath, *Robust techniques for background subtraction in urban traffic video*, SPIE Electronic Imaging, January 2004, pp. 389–404.

[31] Liu S.C., Fu C.W., and Chang S.Y., *Statistical change detection with moments under time-varying illumination*, IP 7 (1998), no. 9, 1258–1268.

[32] Jacob T. Schwartz and Micha Sharir, *Identification of partially obscured objects in two and three dimensions by matching noisy characteristic*, Int. J. Rob. Res. 6 (1987), no. 2, 29–44.

[33] J. Serra, *Image analysis and mathematical morphology*, New York: Academic, 1983.

[34] Raymond van Ee and Barton L. Anderson, *Motion direction, speed and orientation in binocular matching*, Letters to Nature 410 (2001), 690–694.

[35] H. J. Wolfson, *On curve matching*, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990), no. 5, 483–489.

[36] P. L. Worthington and E. R. Hancock, *Object recognition using shape-from-shading*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, 2001, pp. 535–542.

[37] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland, *Pfinder: Real-time tracking of the human body*, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997), no. 7, 780–785.

[38] Alper Yilmaz, Omar Javed, and Mubarak Shah, *Object tracking: A survey*, ACM Comput. Surv. 38 (2006), no. 4, 13.